REMARKS

Prior to this Amendment, claims 1-29 were pending in the application. Claims 1 and 6 have been amended, claims 4, 7, 8 and 14-18 have been canceled, and claims 61-68 have been added without new matter. After entry of this Amendment, claims 1-3, 5, 6, 9-13, 19-29 and 61-68 remain for consideration.

Rejections under 35 U.S.C. §103:

Claims 1-29 were rejected under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 6,948,044 to Chandrasekaran ("Chandrasekaran") in view of U.S. Patent No. 6,718,436 to Kim et al. ("Kim"). This rejection is traversed based on the following remarks.

Introduction:

As discussed in Applicant's background, many businesses and organizations today utilize distributed storage systems and services which allows these businesses and organizations to evolve and expand operations (e.g., globally). While fibre channel switches are conventionally used for individual customers to access dedicated storage systems which may allow protection of proprietary information, these configurations are often costly. Additionally, as storage services are typically rented from vendors which require that such vendors be contacted when certain configuration changes are necessary (e.g., adding storage space, reconfiguration data mappings), maintenance of these storage systems is usually an inconvenience. These problems are only intensified when a business has multiple departments or divisions, each with its own dedicated storage systems (e.g., via a Data Center Manager – DCM) and technology staff. Sharing resources between different departments is another concern. See pages 2 and 3 of Applicant's specification.

The inventive systems disclosed herein can utilize a single network switch system residing in a Storage Area Network (SAN) that allows managing and sharing of resources while securely separating customer data using a storage virtualization process. By providing additional bandwidth and resource connections on demand, the switch system increases the number of host systems that may access the switch system, the number of storage devices providing resources (e.g., storage space), and the number of storage processors that can assist in information virtualization. See switch system 120 in Figure 3 and page 7 of Applicant's specification. The switch system uses a two tier virtualization architecture for managing one or more virtual volumes for a host system: first tier objects assigned to storage processors having connections with one or more storage devices that host virtual volumes for a given virtual volume and second tier objects assigned to storage processors having connections with the host system associated with a given volume. See Figures 6 and 9A-9D of Applicant's specification. These objects allow the switch system to create and manage virtual volumes that are scalable, consistent, and accessible even under abnormal operating conditions. Chandrasekaran and Kim do not disclose or suggest such a network switch system including first and second tiers, and Applicant respectfully requests that the Examiner withdraws the rejection of the claims based on these references for at least this reason.

Independent Claim 1:

Turning to the limitations of independent claim 1, a system for dynamically configuring a virtual volume associated with a host system is disclosed. The system includes a set of storage devices (see storage devices 130 in Figure 1 and ALUs 340, 940 in Figures 3 and 9A-9D which represent "logical units" of one or more storage devices), each of which includes physical block addresses for storing data associated with the virtual volume, and a network switch system connecting the host system and the set of storage devices (see network switch system 120 in Figure 3 and configuration processes performed by network switch system 120 in Figures 9A-9D). The network switch system includes a set of storage processors (e.g., 330 in Figure 3 and 910, 910 in Figures 9A-9D) maintaining virtual volume objects comprising first tier objects (e.g., see T1 layer sub tree 915, 925 in Figure 9A) reflecting a relationship between the physical block addresses and one or more logical partitions of virtual volume data. The system also includes second tier objects (e.g., see T2 layer sub tree 911 in Figure 9A) reflecting a logical configuration of the virtual volume The second tier objects reflect connections between a processor and a host system (e.g., see host 930 in Figure 9A) and the first tier objects reflect connections between a processor and a storage device storing virtual volume data (e.g., see ALU 940 in Figure 9A).

The network switch system uses the first and second tier objects to dynamically update the virtual volume during runtime of the network switch system (e.g., to allow additional host systems to access the same virtual volume, to allow additional storage devices to support the virtual volume). Additionally, the first tier objects have logical connections to both local second tier objects associated with a shared storage processor (e.g., see T1 object 916 and LRN 913 in Figure 9B) and to remote second tier objects associated with at least another storage processor (e.g., see T1 object 916 and RRN 963 in Figure 9B). The first and second tier objects may advantageously include virtual volume definition data that provides a "current view" of the virtual volume (e.g., which host has access, whether striping and/or mirroring is allowed). Also, providing logical connections between the first tier objects and both local and remote second tier objects may advantageously allow different host systems to access the same storage device (e.g., ALU).

Chandrasekaran and Kim fail to disclose or suggest at least the network switch system as recited in independent claim 1 including, inter alia, "a set of storage processors that maintain virtual volume objects comprising first tier objects…and second tier objects… wherein the second tier objects reflect connections between a processor and a host system and the first tier objects reflect connections between a processor and a storage device storing virtual volume data". In contrast, Chandrasekaran is directed to mechanisms that allow a virtual disk address to be efficiently mapped to a particular physical partition in a virtual disk. This reference provides no disclosure of a network switch system having storage processors including first and second tier objects in the manner recited in independent claim 1. More specifically, Chandrasekaran teaches that a variety of indices are used to allow direct access of a physical partition upon identification of a virtual address. See Chandrasekaran at column 1, lines 36-43. In other words, Chandrasekaran is concerned with improving access to data in storage devices as opposed to a system for dynamically configuring a virtual volume.

Turning to Figure 1 of Chandrasekaran, a storage area network includes hosts 111,113, 115 that can perform operations (e.g., reading, writing) with storage devices 121, 123, 153 via switches 101-109. Data striping (i.e., spreading data across multiple storage devices), mirroring (i.e., maintaining data redundantly in multiple storage devices) and concatenation (i.e., combining multiple physical partitions to form a larger virtual disk) can be performed. See column 3, lines 7-67. However, there is nothing unusual about a system whereby hosts have access to virtual address space, a concept which has been known for over 30 years, and is not being claimed by Applicant.

In relation to the recited network switch system including "a set of storage processors that maintain virtual volume objects comprising first tier objects...and second tier objects... wherein the second tier objects reflect connections between a processor and a host system and the first tier objects reflect connections between a processor and a storage device storing virtual volume data," the Examiner cited to Chandrasekaran at column 4, lines 1-56, column 1, line 52 through column 2, line 6, and column 3, lines 26-27. Applicant disagrees that the portions cited by the Examiner disclose or suggest the above recited features of independent claim 1.

Column 4, lines 1-56 generally discuss Figure 2 of Chandrasekaran whereby a switch writes data segments 211-215 in a virtual address space 201 which are striped across virtual disks 241-245 that include physical disk partitions (not shown in Figure 2) and Figure 3 whereby a similar arrangement is provided but with mirroring instead of striping. However, mere disclosure of a generic "switch" that writes data segments in a virtual address space which are striped across virtual disks as in Chandrasekaran does not inherently include "a network switch system including a set of storage processors that maintain virtual volume objects comprising first tier objects (e.g., virtual volume definitions)...and second tier objects (e.g., virtual volume definitions)... wherein the second tier objects reflect connections between a processor and a host system and the first tier objects reflect connections between a processor and a storage device storing virtual volume data" as recited in independent claim 1. More specifically, the generic switch of Chandraskaran is merely associated with having data written to a particular physical partition associated with a particular virtual address and just provides no disclosure of a set of storage processors having first and second tier virtual volume objects.

As called out in independent claim 1, the set of storage processors having first and second tier virtual volume objects in the network switch system advantageously allows the dedicated storage system(s) of multiple departments of one or more companies to be configured (e.g., change/add hosts and/or storage systems that are associated with a particular virtual volume) by the same network switch system because the processors of the network switch system include the first and second tier objects having current volume definitions (e.g., which host(s) has access, whether striping and/or mirroring is allowed) which allows efficient dynamic reconfiguration of virtual volumes. Here, the Examiner has cited to no portion of Chandrasekaran that discloses first and second tier objects reflecting relationships between storage devices and a processors of a network switch system, and host system and processors of a network switch system, respectively. Rather, the portions cited by the Examiner only appear to discuss writing and reading of data onto virtual disks involving striping, mirroring and concatenation which is distinct from the recited network switch system which allows for *dynamic updating* of virtual volumes during runtime of a network switch system.

Even if Chandrasekaran did somehow disclose first and second tier objects in the manner disclosed in independent claim 1, there is no disclosure of "a network switch system including a set of storage processors that maintain virtual volume objects comprising first tier objects and second tier objects in the manner described in independent claim 1." That is, Chandrasekaran does not disclose storage processors of a switch system having the first and second tier objects. This placement of the first and second tier objects advantageously allows a single network switch system to manager/configure virtual volumes of, for instance, a large number of organizational departments.

The next portion cited by the Examiner (column 1, line 52 through column 2, line 6 of Chandrasekaran) generally discusses a fibre channel switch including an interface and a processor – the interface receives a disk access request from a host including a virtual disk address, and the processor identifies an entry in an index corresponding to a physical partition based on the virtual disk address. Again however, this portion merely discusses a switch that receives a disk access request from a host at an interface and a processor that locates a particular physical partition for reading or writing data instead of *a switch system including a set of storage processors that maintain virtual volume objects comprising first tier objects and second tier objects* in the manner described in independent claim 1. More particularly, this generic "switch" of Chandrasekaran would not be able to somehow dynamically update a virtual volume during runtime of a network switch system (e.g., by restructuring a logical tree including first and second tier objects) as in independent claim 1.

The other portion cited by the Examiner (column 3, lines 26-27 of Chandrasekaran) merely mentions that virtualization and mapping are used to spread data across multiple storage devices. But again, this is not surprising as these concepts have been used to spread data across multiple storage devices for many years, and has nothing to do with a switch system including a set of storage processors that maintain virtual volume objects comprising first tier objects and second tier objects in the manner described in independent claim 1. This portion does not disclose an arrangement that would somehow be able to dynamically update a virtual volume during runtime of a network switch system as in independent claim 1 (i.e., "virtualization" and "mapping" do not inherently operate to dynamically update a virtual volume during runtime of a network switch system that includes a set of processors with first and second tier virtual volume objects).

As Chandrasekaran fails to disclose a number of limitations from independent claim 1, Chandrasekaran cannot be used to sustain a rejection of independent claim 1. Further, Kim was not cited for any teachings relevant to the above recited features of independent claim 1. Applicant therefore respectfully requests that independent claim 1 be indicated allowable.

Chandrasekaran and Kim also fail to teach "wherein the network switch system uses the first and second tier objects to dynamically update the virtual volume during runtime of the network switch system" as recited in independent claim 1. After admitting that Chandrasekaran is deficient in this regard (see bottom of page 2 of Final Office Action), the Examiner asserted that Kim discloses "the use of different tiers of virtual volume objects used to update the virtual volume during runtime," and that it would have been obvious to disclose the specific connectedness of the tier objects to dynamically configure the virtual volume because "these objects necessarily must interact with each other (and must thus have connections) for them to be used in the configuration process." Applicant disagrees that the Examiner's combination discloses the above recited feature of independent claim 1.

Kim is directed to a method for managing a logical volume that minimizes a size of metadata and that supports dynamic online resizing including a) gathering disk partitions (i.e., physical partitions, see column 6, line 58) in response to a request for creating a logical volume, b) generating metadata including disk partitions forming the logical volume and storing the metadata in the disk partitions, c) resizing the logical volume and modifying the metadata, and d) calculating and returning a physical address corresponding to a logical address of the logical volume. See Kim at column 4, lines 7-54 (cited by Examiner on page 3 of Final Office Action) and Figures 1 and 3. A metadata table 70 (see Figure 3), including information regarding which disk/physical partitions make up which logical volume, is stored in each disk/physical partition (Emphasis added). The metadata table 70 includes a physical partition map 71 and a logical volume map 72, among other maps (shown in Figures 4-7). See column 7, lines 7-24 (cited by Examiner on page 3 of Final Office Action).

For instance, the physical partition (PP) map 71 of the metadata table 70 (shown in Figure 4 of Kim) includes a Volume_ID 82 for a logical volume formed by the particular disk partition that that metadata table 70 is stored in, and a Physical_Partition_ID 83 for identifying itself in the logical volume. See column 8, lines 9-25 of Kim. Thus, read in the best light, the metadata table 70 of Kim provides information representing a relationship between logical volumes and physical partitions, and it may be used to dynamically "resize" a logical volume. However, independent claim 1 recites "the network switch system uses the first and second tier objects to dynamically update the virtual volume during runtime of the network switch system" (Emphasis added).

Thus, while Kim discloses dynamic resizing of a logical volume, Kim does not disclose doing so using first and second tier objects in the manner recited in independent claim 1. More specifically, and even assuming the various "maps" of the metadata table 70 are somehow equivalent to the recited "tier objects", there is no support in Kim for some of the maps "reflecting connections between a processor and a host system" and some of the maps "reflecting connections between a processor and a storage device storing virtual volume data" as in independent claim 1. As discussed previously, the first and second tier objects of independent claim 1 can be used to dynamically adjust, for instance, which host system(s) has or have access to a particular virtual volume, and/or which storage device is associated with a particular storage process of the network switch system. In contrast, the metadata table 70 (and its various maps) appears to only be concerned with relationships between a logical volume and physical partitions (e.g., which physical partitions make up a logical volume, how the physical partitions are associated with mirroring and/or striping). Thus, when a logical volume in Kim is "dynamically resized," this is used in the context of updating the metadata table 70 instead of using first and second tier objects to, for instance, define which host system(s) has or have access to a particular virtual volume, and/or which storage device is associated with a particular storage process of the network switch system. Due to this failure of Chandrasekaran and Kim to disclose a "network switch system (that) uses first and second tier objects to dynamically update a virtual volume during runtime of the network switch system" as recited in independent claim 1, Applicant respectfully requests that this claim be indicated allowable.

Additionally, Kim provides absolutely no disclosure of "a network switch system…including a set of processors…maintaining…first and second tier objects" as recited in independent claim 1. *Rather*, the metadata table 70 is stored *in the disk/physical partitions* (see Kim at column 7, lines 18-20) as opposed to *in processors of a network switch system* as recited in independent claim 1. This placement of the first and second tiers allows a single network switch system to manager/configure virtual volumes of, for instance, a large number of organizational departments. Thus, even assuming that Chandrasekaran and Kim were properly combinable, the references still do not disclose the system of independent claim 1 due to Kim's failure to disclose "wherein the network switch system uses the virtual volume objects (in the manner recited in independent claim 1) to dynamically update the virtual volume during runtime of the network switch system." Because of this additional failure of the references to disclose a limitation of independent claim 1, Applicant respectfully requests that independent claim 1 be indicated allowable.

Chandrasekaran and Kim also fail to teach "the first tier objects have logical connections to both local second tier objects associated with a shared storage processor and to remote second tier objects associated with at least another storage processor" as recited in independent claim 1. After admitting, again, that Chandrasekaran is deficient in this regard (see bottom of page 2 of Final Office Action), the Examiner asserted that Kim discloses "the use of different tiers of virtual volume objects used to update the virtual volume during runtime," and that it would have been obvious to disclose the specific connectedness of the tier objects to dynamically configure the virtual volume because "these objects necessarily must interact with each other (and must thus have connections) for them to be used in the configuration process." Applicant disagrees that the Examiner's combination discloses the above recited feature of independent claim 1.

Again, it appears the Examiner has equated the recited first and second tier objects to the metadata table 70 and/or the various maps of the metadata table 70 of Kim. However, and as discussed above, the metadata table 70 is stored *in each of the disk/physical partitions* as opposed to *in storage processors of a network switch system* as in independent claim 1. Thus, the metadata table 70 cannot possibly have first tier objects with *logical connections* to both local second tier objects associated with a *shared storage processor* and to remote second tier objects associated with at least *another storage processor* because the metadata table 70 is not associated with any storage processors of a network switch system in the first place. These connections advantageously allow, for instance, a host system associated with one processor of a network switch system to access a storage device associated with another processor of the network switch system.

Any combination of Chandrasekaran and Kim would result in a system whereby some sort of switch would write data segments in a virtual address space comprised of a number of virtual disks, and each of the virtual disks would be made up of physical disk partitions having metadata tables with maps representing relationships between the virtual address space and disks and the physical disk partitions. Such a system does not include, inter alia, a network switch system including a set of storage processors having first and second tier objects that may be used to dynamically update a virtual volume during runtime of the network switch system as recited in independent claim 1. Due to these additional failures of Chandrasekaran and Kim to disclose limitations of independent claim 1, Applicant respectfully requests that independent claim 1 be indicated allowable.

Claims 2, 3, 5-8, 11-13 and 19-29 depend from independent claim 1 and are believed allowable over Chandrasekaran and Kim for at least the reasons for allowing independent claim 1 over these references. Further, these claims recite additional features not disclosed or suggested by the references. For instance, claim 3 recites "wherein the network switch system dynamically updates the virtual volume by at least one of adding (see Figure 9B of Applicant's specification) a virtual volume object to a storage processor, removing (see Figure 9D) a virtual volume object from a storage processor, and moving (see Figure 9C) a virtual volume object from one storage processor to another storage processor." The Examiner cited Chandrasekaran at Figure 9, column 1, lines 35-67 and column 5, lines 3-66. However, these portions generally disclose, as discussed previously, providing direct access to a physical partition upon identification of a virtual address instead of dynamically updating a virtual volume by adding, removing or moving virtual volume objects (e.g., the recited first and second tier objects) within or between storage processors of a network switch system as in claim 5. The Examiner also cited Kim at column 4, lines 7-54 and column 7, lines 7-24. Again, however, these portions merely disclose a metadata table stored within each disk/physical partition that stores various maps and provides no disclosure of adding, removing or moving virtual volume objects (e.g., the recited first and second tier objects) within or between storage processors of a network switch system as in claim 5. Thus, claim 3 is believe allowable over Chandrasekaran and Kim for these additional reasons.

New Claims:

Independent Claim 61:

Independent claim 61 is directed to a system for dynamically configuring a virtual volume associated with a host system. The system includes a set of storage devices, each of which includes physical block addresses for storing data associated with the virtual volume, and a network switch system connecting the host system and the set of storage devices. The network switch system includes a set of storage processors maintaining virtual volume objects comprising first tier objects and second tier objects each being associated with different types of volume management processes. The second tier objects reflect connections between a processor and a host system and the first tier objects reflect connections between a processor and a storage device storing virtual volume data. The network switch system uses the virtual volume objects to dynamically update the virtual volume during runtime of the network switch system.

The network switch system also includes a Virtualization Block (VB) component (e.g., VBM 338 in Figure 4 of Applicant's specification) that, based on a host system request, restructures a logical tree reflecting relationships between the second tier and first tier objects of the virtual volume (see ¶ 049 and 0109 of Applicant's specification). More specifically, when the host system request requires the VB component to add a new second tier object (e.g., T2 layer sub tree 960 in Figure 9B) to a target storage processor (e.g., SP 920) that is logically related to a first tier object (e.g., T1 layer sub tree 925, 915), the VB component configures the new second tier object to include a reference node (e.g., RRN 963, LRN 962) that references the first tier object. This advantageously allows a new host (e.g., host 930) to have access to a virtual volume including one or more storage devices (e.g., ALU 940, 950). As discussed previously, this arrangement would be useful for centrally managing the dedicated storage devices or virtual volumes of multiple departments or divisions of one or more organizations.

Chandrasekaran and Kim do not disclose or suggest the system of independent claim 61 because Chandrasekaran and Kim do not disclose or suggest a number of limitations of independent claim 61. As independent claim 61 incorporates subject matter from claims 7 and 8, Applicant will also address the Examiner's arguments in relation to these claims.

As substantially discussed above in relation to independent claim 1, Chandrasekaran and Kim do not disclose or suggest at least "a set of storage processors that maintain virtual volume objects comprising first tier objects…and second tier objects… wherein the second tier objects reflect connections between a processor and a host system and the first tier objects reflect connections between a processor and a storage device storing virtual volume data." In contrast, the portions of Chandrasekaran cited by the Examiner generally discuss a switch that receives a disk access request from a host at an interface and a processor that locates a particular physical partition for reading or writing data, and does not disclose *a switch system including a set of storage processors that maintain virtual volume objects comprising first tier objects and second tier objects* in the manner described in independent claim 1.

As discussed throughout, this arrangement allows a single network switch system to centrally manage/update/configure the dedicated storage devices of a plurality of organizations and/or departments by way of maintaining a "current view" of connections between processors of the switch system and host systems and storage devices. The generic "switch" of Chandrasekaran would not be able to somehow dynamically update a virtual volume during runtime of a network switch system (e.g., by restructuring a logical tree including first and second tier objects) as in independent claim 1. Additionally, Kim was not cited for any teachings relevant to the above recited features of independent claim 1. Due to at least these deficiencies, Applicant respectfully requests that independent claim 61 be indicated allowable.

As also discussed above in relation to independent claim 1, Chandrasekaran and Kim also do not disclose or suggest at least "wherein the network switch system uses the first and second tier objects to dynamically update the virtual volume during runtime of the network switch system." The Examiner admitted that Chandrasekaran is deficient in this regard (see bottom of page 2 of Final Office Action), and the metadata table 70 of Kim does not include second tier objects that reflect connections between a processor and a host system and first tier objects that reflect connections between a processor and a storage device storing virtual volume data as recited in independent claim 61. For these additional reasons, Applicant respectfully requests that independent claim 61 be indicated allowable.

As additionally discussed in relation to independent claim 1, Chandrasekaran and Kim also do not disclose or suggest at least "a network switch system...including a set of processors...maintaining...first and second tier objects." In contrast, the generic "switch" of Chandrasekaran is not disclosed as including a set of processors with first and second tier objects, and the metadata table 70 of Kim is actually stored in the *disk/physical partitions* as opposed to *in processors of a network switch system* as recited in independent claim 61. This placement of the first and second tiers allows a single network switch system to manager/configure virtual volumes of, for instance, a large number of organizational departments. For these additional reasons, Applicant respectfully requests that independent claim 61 be indicated allowable.

Additionally, Chandrasekaran and Kim do not disclose "a Virtualization Block (VB) component that, based on a host system request, restructures a logical tree reflecting relationships between the second tier and first tier objects of the virtual volume, wherein when the host system request requires the VB component to add a new second tier object to a target storage processor that is logically related to a first tier object, the VB component configures the new second tier object to include a reference node that references the first tier object" as recited in independent claim 61.

In relation to now canceled claims 7 and 8, the Examiner cited to Chandrasekaran at Figure 9, column 1, lines 35-67, column 5, lines 3-66, and column 4, lines 1-56. However, and as discussed earlier, these portions merely generally discuss a switch that receives a disk access request from a host at an interface and a processor that locates a particular physical partition for reading or writing data (whereby direct access to the physical partition is provided upon identification of a virtual address in the request) instead of any sort of component that configures a new second tier object in a logical tree to include a node that references a logically related first tier object. That is, mere disclosure of a generic "switch" that writes data segments in a virtual address space which are striped across virtual disks as in Chandrasekaran has nothing to do with objects that define or reflect relationships between storage processors of a network switch system, and host systems and storage devices as in independent claim 61.

The Examiner also cited to Kim at column 4, lines 7-54 and column 7, lines 7-24. However, these portions merely disclose a metadata table (stored within disk/physical partitions) that stores various maps defining relationships between logical volumes and physical partitions instead of any sort of component that configures a new second tier object in a logical tree to include a node that references a logically related first tier object. As discussed previously, Kim does not even disclose first tier objects reflecting connections between a processor and a host system and second tier objects reflecting connections between a processor and a storage device storing virtual volume data, much less within storage processors of a network switch system as recited in independent claim 61.

Due to all of the above failures of Chandrasekaran and Kim to disclose or suggest various limitations of independent claim 61, Applicant respectfully requests that this claim be indicated allowable. Additionally, claims 62 and 63 depend from independent claim 61 and are believed allowable over Chandrasekaran and Kim for at least the reasons for allowing independent claim 61 over these references.

Independent Claim 64:

Independent claim 64 is directed to a system for dynamically configuring a virtual volume associated with a host system. The system includes a set of storage devices, each of which includes physical block addresses for storing data associated with the virtual volume, and a network switch system connecting the host system and the set of storage devices. The network switch system includes a set of storage processors maintaining virtual volume objects comprising first tier objects and second tier objects each being associated with different types of volume management processes. The second tier objects reflect connections between a processor and a host system and the first tier objects reflect connections between a processor and a storage device storing virtual volume data. The network switch system uses the virtual volume objects to dynamically update the virtual volume during runtime of the network switch system.

The network switch system also includes a Virtualization Block (VB) component (e.g., VBM 338 in Figure 4 of Applicant's specification) that, based on a host system request, restructures a logical tree reflecting relationships between the second tier and first tier objects of the virtual volume (see ¶ 049 and 0109 of Applicant's specification). Additionally, the network switch system includes a Virtualization Coherency Manager (VCM) (e.g., VCM 337 in Figure 3 and VCM 336 in Figures 4, 7A, 7B) that assigns the first tier objects to selective ones of the storage processor and the second tier objects to selective ones of the second tier storage processors based on the restructured logical tree. When the host system request requires the VB component to move an existing first tier object (e.g., T1 layer sub tree 926 in Figure 9A) from a first storage processor (e.g., SP 920) to a second processor (e.g., SP 910) having a remote reference (e.g., RRN 914) to the existing first tier object, the VCM sends a new second tier object tree to the first and second storage processors that removes any references to the existing first tier object. See Figure 9C and ¶ 0112-0113 of Applicant's specification. This arrangement may be useful to create an updated logical tree for a virtual volume reflecting that a particular storage device is no longer associated with a particular virtual volume.

Chandrasekaran and Kim do not disclose or suggest the system of independent claim 64 because Chandrasekaran and Kim do not disclose or suggest a number of limitations of independent claim 64. As independent claim 64 incorporates subject matter from claim 14, Applicant will also address the Examiner's arguments in relation to this claim.

As substantially discussed above in relation to independent claims 1 and 61, Chandrasekaran and Kim do not disclose or suggest at least "a set of storage processors that maintain virtual volume objects comprising first tier objects...and second tier objects... wherein the second tier objects reflect connections between a processor and a host system and the first tier objects reflect connections between a processor and a storage device storing virtual volume data," "wherein the network switch system uses the first and second tier objects to dynamically update the virtual volume during runtime of the network switch system," and "a network switch system...including a set of processors...maintaining...first and second tier objects."

Due to at least these deficiencies, Applicant respectfully requests that independent claim 64 be indicated allowable.

In relation to now canceled claim 14, the Examiner cited to Chandrasekaran at Figure 9, column 1, lines 35-67 and column 5, lines 3-66. However, and as discussed earlier, these portions merely generally discuss a switch that receives a disk access request from a host at an interface and a processor that locates a particular physical partition for reading or writing data (whereby direct access to the physical partition is provided upon identification of a virtual address in the request) instead of any sort of component that sends a new second tier object tree to first and second storage processors (of a network switch system) that removes any references to an existing first tier object when a host system request requires the component to move an existing first tier object from a first storage processor to a second processor having a remote reference to the existing first tier object.

The Examiner also cited to Kim at column 4, lines 7-54 and column 7, lines 7-24. However, these portions merely disclose a metadata table (stored within disk/physical partitions) that stores various maps defining relationships between logical volumes and physical partitions instead of any sort of component that sends a new second tier object tree to first and second storage processors (of a network switch system) that removes any references to an existing first tier object. As discussed previously, Kim does not even disclose first tier objects reflecting connections between a processor and a host system and second tier objects reflecting connections between a processor and a storage device storing virtual volume data, much less within storage processors of a network switch system as recited in independent claim 64.

Due to all of the above failures of Chandrasekaran and Kim to disclose or suggest various limitations of independent claim 64, Applicant respectfully requests that this claim be indicated allowable. Additionally, claims 65-68 depend from independent claim 64 and are believed allowable over Chandrasekaran and Kim for at least the reasons for allowing independent claim 64 over these references.

Conclusions

Based upon the foregoing, Applicant believes that all pending claims are in condition for allowance and such disposition is respectfully requested. In the event that a telephone conversation would further prosecution and/or expedite allowance, the Examiner is invited to contact the undersigned.

No fee is believed due with this Amendment. However, please credit any overpayment or charge any underpayment to Deposit Account No. 50-1419.

Respectfully submitted,

MARSH FISCHMANN & BREYFOGLE LLP

Date: MAY 13, 2010

By: ____________________
Jonathon A. Szumny
Reg. No. 57,695
Tel: 303-486-6307